

SISTEM PELACAKAN DAN KEAMANAN KENDARAAN BERMOTOR SECARA REAL-TIME BERBASIS NODEMCU ESP8266 DENGAN APLIKASI MOBILE

Pradika Candra Aditya^{1*)}, Djuniadi Djuniadi¹, dan Abdurrahman Hamid Al-azhari¹

¹Departemen Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang, Semarang, Indonesia

^{*)E-mail : xaditya242@students.ac.unnes.ac.id}

Abstrak

Sistem pelacakan kendaraan ini dirancang untuk memudahkan pemilik dalam memantau lokasi kendaraan secara *real-time* serta meningkatkan keamanannya melalui aplikasi *mobile*. Teknologi GPS dan WiFi digunakan untuk mengirimkan data lokasi dalam bentuk koordinat, jarak, serta informasi lain. Modul GPS Neo 6M memperoleh data lokasi yang diproses oleh NodeMCU ESP8266 dan dikirim ke *database* melalui WiFi. Aplikasi *mobile* menampilkan peta, alamat, kecepatan kendaraan, kekuatan sinyal, serta riwayat pergerakan selama tiga hari terakhir dengan Google Maps API. Penelitian ini mencakup perancangan perangkat keras dan lunak serta pengujian akurasi pelacakan, efektivitas keamanan, dan kinerja sistem. Hasil menunjukkan sistem dapat menampilkan lokasi kendaraan dengan akurasi bervariasi dan mencatat riwayat pergerakan dengan baik. Meskipun terdapat latensi rata-rata 1423 ms dalam pengiriman data, fitur keamanan menunjukkan tingkat keberhasilan 100% dengan respons cepat. Dengan demikian, sistem ini efektif untuk pemantauan dan perlindungan kendaraan, meskipun masih diperlukan optimasi lebih lanjut untuk meningkatkan akurasi dan kecepatan respons.

Kata kunci: pelacakan kendaraan, aplikasi mobile, mikrokontroler NodeMCU ESP8266, teknologi WiFi dan GPS, riwayat pergerakan kendaraan, Google Maps.

Abstract

This vehicle tracking system is designed to help owners monitor their vehicle's location in real time and enhance security through a mobile application. GPS and WiFi technology are used to transmit location data in the form of coordinates, distance, and other information. The GPS Neo 6M module obtains location data, which is processed by the NodeMCU ESP8266 and sent to a database via WiFi. The mobile application displays maps, addresses, vehicle speed, signal strength, and a three-day movement history using the Google Maps API. This research includes the design of hardware and software, as well as testing for tracking accuracy, security effectiveness, and system performance. Results show that the system successfully displays vehicle locations with varying accuracy and records movement history efficiently. Although there is an average latency of 1423 ms in data transmission, the security feature achieves a 100% success rate with a quick response time. Thus, this system is effective for vehicle monitoring and protection, though further optimization is needed to improve accuracy and response speed.

Keywords: vehicle tracker, mobile application, NodeMCU ESP8266 microcontroller, WiFi and GPS technology, vehicle movement history, Google Maps.

1. Pendahuluan

Pertumbuhan populasi penduduk yang semakin meningkat berbanding lurus dengan jumlah kepemilikan kendaraan pribadi baik sepeda motor maupun mobil. Berdasarkan data statistik perkembangan jumlah kendaraan bermotor, pada tahun 2022 mengalami peningkatan sebanyak 5.263.034 unit dari tahun sebelumnya untuk jenis kendaraan sepeda motor[1]. Namun, banyaknya jumlah kendaraan bermotor ternyata juga sejalan dengan

peningkatan kasus pencurian kendaraan bermotor di Indonesia. Berdasarkan data yang dilaporkan oleh Badan Pusat Statistik (BPS) terdapat 372.965 kejahatan yang terjadi di Indonesia sepanjang 2022. Dari jumlah itu, sebanyak 14.184 kejahatan berupa pencurian kendaraan bermotor[2]. Dalam kasus pencurian kendaraan bermotor, sering kali pemilik kesulitan dalam melacak kendaraan yang dibawa oleh pelaku. Oleh karena itu, dibutuhkan sistem yang dapat melakukan pelacakan dan pemantauan

lokasi kendaraan secara *real-time* untuk meningkatkan keamanan kendaraan bermotor.

Beberapa penelitian sebelumnya telah mengusulkan solusi dalam bidang pelacakan kendaraan. Misalnya, penelitian tentang “Keamanan Kendaraan untuk Melacak Sepeda Motor yang Hilang dengan menggunakan GPS Berbasis Smartphone”[3] yang menggunakan metode pengiriman lokasi melalui SMS dan Google Maps. Namun, pendekatan ini memiliki keterbatasan karena SMS merupakan teknologi komunikasi lama dengan keterbatasan dalam pengiriman data dan tidak memiliki fitur interaktif[4]. Penelitian lain juga dilakukan dengan “Implementasi Internet of Things pada Sistem Informasi Pelacakan Kendaraan Bermotor Menggunakan GPS Berbasis Web”[5] namun dikarenakan pada penelitian tersebut sistem informasi berbasis *web* sehingga masih menjadi kekurangan untuk kemudian dikembangkan menjadi sistem informasi berbasis *mobile*. Penelitian lain juga sudah dilakukan dengan “Rancang Bangun Alat Pelacak Posisi Kendaraan Berbasis IoT”[6], telah mengintegrasikan sistem keamanan dengan fitur pemutusan arus kontak kendaraan, tetapi pengguna masih harus menggunakan aplikasi tambahan, yang mengurangi efisiensi sistem secara keseluruhan.

Berdasarkan kelemahan dari penelitian terdahulu, penelitian ini berkontribusi dalam pengembangan Sistem Pelacakan Kendaraan Bermotor secara Real-Time dan Pemantauan Riwayat Lokasi berbasis NodeMCU ESP8266 dengan Aplikasi Mobile. ESP8266 dipilih karena memiliki modul WiFi terintegrasi, konsumsi daya rendah, dan harga terjangkau. Sistem ini menggunakan GPS untuk memperoleh lokasi kendaraan dan mengirimkannya ke *database* melalui WiFi. Aplikasi mobile dikembangkan untuk menampilkan lokasi kendaraan secara *real-time* serta menyediakan fitur tambahan, termasuk sistem pengaman yang memungkinkan pengguna memutus arus kunci kontak dari jarak jauh. Dengan solusi ini, pemilik kendaraan dapat lebih mudah memantau kendaraannya serta meningkatkan keamanan guna mencegah dan menangani potensi pencurian.

2. Metode

Metode pengembangan sistem pelacakan kendaraan bermotor dan pemantauan riwayat lokasi kendaraan dilakukan melalui rancangan arsitektur sistem, rancangan perangkat sistem, dan pengujian sistem.

2.1. Rancangan Arsitektur Sistem

Sistem yang akan dibuat dalam penelitian ini merupakan salah satu implementasi dari *Internet of Things* (IoT). IoT merupakan sistem yang menghubungkan perangkat komputer, mesin mekanis dan *digital*, objek, atau individu yang dilengkapi dengan pengenal unik. Sistem ini

memungkinkan pengiriman data melalui jaringan tanpa memerlukan interaksi langsung antara manusia dengan manusia atau manusia dengan komputer[7]. Secara umum arsitektur dari IoT memiliki tiga bagian yaitu *Perception Layer*, *Network Layer*, dan *Application Layer*[8].

2.1.1. Perception Layer

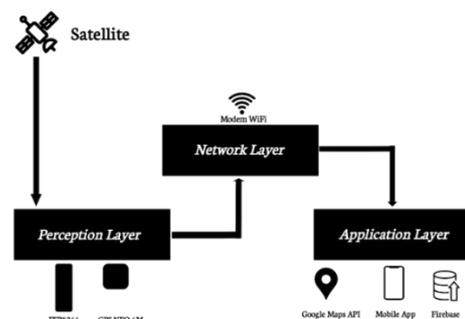
Perception layer atau sensor layer berinteraksi langsung dengan sensor dan aktuator untuk mengumpulkan serta memproses data secara *real-time* sebelum dikirim ke lapisan berikutnya[9]. Pada penelitian ini, perangkat keras yang digunakan meliputi mikrokontroler NodeMCU ESP8266 sebagai pusat pengolahan data, Modul GPS NEO 6M untuk memperoleh data lokasi, serta komponen pendukung lainnya untuk mengoptimalkan kinerja sistem.

2.1.2. Network Layer

Lapisan yang berada di tengah ini bisa juga disebut sebagai *communication layer* yang merupakan inti dari ekosistem IoT[10]. Lapisan ini bertanggung jawab untuk mengelola konektivitas jaringan, pengiriman pesan antar *server*, penghubung antara lapisan persepsi dan lapisan di atasnya, menyediakan jalur sehingga data dapat ditransfer ke *server* IoT, aplikasi dan perangkat melalui jaringan yang saling terhubung[11]. Dalam penelitian ini jaringan yang digunakan untuk melakukan transmisi data ke *database* adalah jaringan WiFi. Jaringan WiFi akan membantu perangkat pada *perception layer* untuk mentransmisikan data menuju *database server*.

2.1.3. Application Layer

Lapisan ini berada pada bagian teratas kerangka IoT dan berfungsi sebagai antarmuka antara pengguna dan aplikasi. *Application layer* menerima data dari lapisan sebelumnya lalu diolah untuk operasi tertentu[12]. Dalam penelitian ini, informasi didapatkan dari *database* Firebase sebagai penyimpanan data lokasi dan informasi lainnya yang telah diterima dari perangkat *perception layer*, kemudian akan diproses menggunakan Google Maps API dan beberapa proses lainnya untuk dapat ditampilkan pada aplikasi *mobile* sehingga lebih mudah, praktis, dan interaktif dari sisi pengguna. Arsitektur IoT sistem pelacakan kendaraan ditunjukkan pada Gambar 1

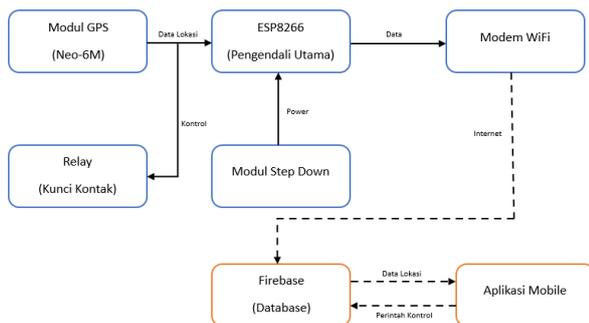


Gambar 1. Arsitektur IoT sistem pelacakan kendaraan

Pada arsitektur yang ditunjukkan Gambar 1, perangkat GPS memperoleh data koordinat *latitude* dan *longitude* dari satelit melalui modul GPS NEO 6M. Data tersebut kemudian dikirim ke Firebase menggunakan koneksi internet yang disediakan oleh modem WiFi. Firebase akan menerima data dan akan mengirimkannya ke perangkat aplikasi untuk kemudian diolah. Hasil pengolahan data ini ditampilkan pada sistem informasi berbasis aplikasi *mobile* untuk memudahkan pengguna dalam melacak kendaraan bermotor.

2.2. Rancangan Perangkat Sistem

Untuk mendukung implementasi sistem yang dirancang, diperlukan perangkat keras yang berfungsi sebagai *platform* fisik untuk menjalankan sistem IoT. Pada bagian ini, akan dijelaskan kebutuhan dan spesifikasi komponen yang digunakan, serta proses desain dan pembuatan perangkat keras yang melibatkan perakitan komponen menjadi sebuah sistem yang berfungsi sesuai rancangan. Diagram blok sistem ditunjukkan pada Gambar 2



Gambar 2. Diagram blok sistem

Sebagaimana ditunjukkan pada Gambar 2, sistem terdiri dari modul GPS, ESP8266, modem WiFi, Firebase, dan aplikasi mobile. GPS memperoleh data lokasi, ESP8266 memproses dan mengirimkannya ke Firebase, lalu aplikasi menampilkan posisi kendaraan secara *real-time*. Sistem juga dilengkapi relay sebagai kunci kontak yang dikontrol melalui aplikasi serta modul *step-down* untuk menyesuaikan tegangan.

2.2.1. Kebutuhan dan Spesifikasi Komponen

a. NodeMCU ESP8266

NodeMCU merupakan *platform* IoT yang bersifat *opensource* yang terdiri dari perangkat keras berupa *system on chip* ESP826. NodeMCU bisa dianalogikan sebagai sebuah mikrokontroler seperti Arduino, yang ditambah dengan modul WiFi ESP8266[13]. Sistem WiFi tersebut menjadi pembeda antara ESP8266 dengan mikrokontroler lainnya, karena dapat menjadi alat *controlling* sekaligus *monitoring* jarak jauh.[14] Secara *default*, istilah NodeMCU lebih merujuk pada *firmware* dibandingkan perangkat kerasnya[15]. NodeMCU ESP8266 digunakan

pada penelitian ini dikarenakan kemampuannya yang dapat terhubung ke jaringan internet tanpa adanya modul tambahan. NodeMCU ESP8266 akan digunakan sebagai pusat memproses data yang diterima oleh sensor.

b. Modul GPS Neo 6M

GPS (*Global Positioning System*) merupakan sistem satelit navigasi global yang menyediakan informasi lokasi geografis dan waktu ke penerima GPS dimana saja di Bumi[16]. NEO-6M adalah modul penerima sinyal GPS yang dikembangkan oleh U-Blox. Modul ini terdiri dari komponen GY-GPSV3-NEO, biasanya digunakan dalam berbagai aplikasi termasuk pelacakan kendaraan. PCB pada modul ini berisi chipset GPS NEO-6M, antenna, dan regulator tegangan. Chipset ini dapat menerima sinyal dari beberapa satelit GPS secara bersamaan dengan menggunakan algoritma tertentu untuk menghitung posisi, kecepatan, dan waktu perangkat secara akurat[17].

c. Modem WiFi

Perangkat yang dihasilkan dalam penelitian ini bersifat *mobile*, sehingga membutuhkan koneksi internet independen untuk memastikan pengiriman data yang stabil. Mengingat keterbatasan akses WiFi eksternal dalam sistem pelacakan kendaraan, modem WiFi berbasis jaringan seluler digunakan sebagai penghubung antara jaringan internet seluler dan perangkat IoT atau sistem lain yang memerlukan koneksi internet[18].

d. Modul Step Down

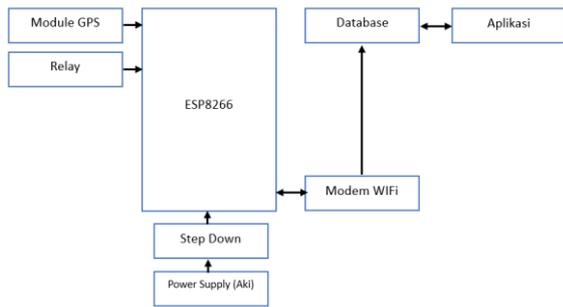
Modul step down DC MP1584 digunakan untuk mengatasi perbedaan antara tegangan yang dibutuhkan dengan tegangan yang tersedia. Dalam penelitian ini, sumber tegangan diambil dari aki sepeda motor dengan tegangan 12 volt, tegangan ini kemudian diatur oleh modul step-down MP1584 untuk memberikan tegangan tetap 5 volt [19] yang dibutuhkan oleh NodeMCU ESP8266.

e. Komponen Pendukung

Dalam perancangan perangkat keras, beberapa komponen pendukung dibutuhkan untuk mendukung kinerja dan konektivitas secara keseluruhan. Komponen-komponen ini berfungsi untuk memastikan konektivitas, kestabilan aliran daya, dan sinyal antar bagian sistem sehingga dapat beroperasi secara optimal dan efisien.

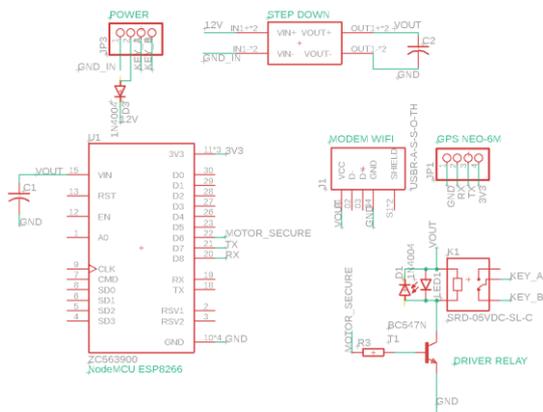
2.2.2. Desain dan Pembuatan Perangkat Keras

Pembuatan perangkat keras membutuhkan beberapa komponen sebagaimana disebutkan sebelumnya. Diagram Blok Perangkat Keras ditunjukkan pada Gambar 3



Gambar 3. Diagram Blok Perangkat Keras

Diagram blok pada Gambar 3 menunjukkan perancangan perangkat keras sistem pelacakan kendaraan berbasis IoT. Sistem ini menggunakan ESP8266 sebagai pengendali utama yang menerima data lokasi dari modul GPS dan mengontrol relay sebagai fitur keamanan. Modul *step-down* digunakan untuk menyesuaikan tegangan dari sumber daya (aki) agar sesuai dengan kebutuhan ESP8266 dan komponen lainnya. Data lokasi kemudian dikirimkan melalui modem WiFi ke *database*, yang dapat diakses oleh aplikasi *mobile* untuk pemantauan *real-time*. Rancangan Perangkat Keras ditunjukkan pada Gambar 4

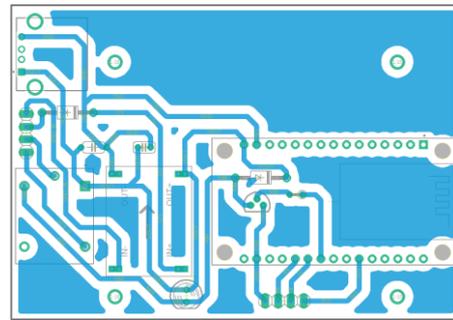


Gambar 4. Rancangan Perangkat Keras

Port I/O yang digunakan dalam projek ini adalah sebagai berikut:

1. Pin D8 terhubung ke pin TX GPS
2. Pin D7 terhubung ke pin RX GPS
3. Pin D3 terhubung ke pin Trigger Kunci Kontak

Komponen-komponen sebagaimana pada Gambar 4 dirangkai dalam satu kesatuan menggunakan PCB. Penggunaan PCB bertujuan untuk menyederhanakan rangkaian, meningkatkan ergonomis alat, serta memastikan efisiensi penggunaan ruang tanpa mengurangi fungsionalitas. Rangkaian PCB ini dirancang menggunakan perangkat lunak Cadsoft EAGLE sehingga menjadi seperti pada Gambar 5.



Gambar 5. Layout PCB

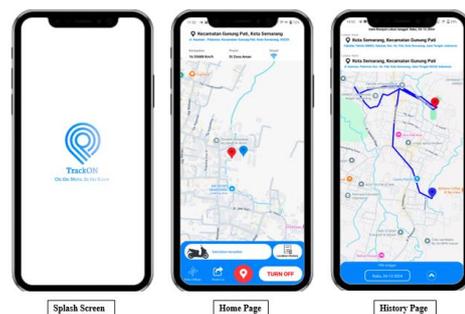
Untuk menjaga keamanan rangkaian dan melindungi komponen dari pengaruh lingkungan luar, alat ini dilengkapi dengan kotak hitam sebagai pelindung untuk menjaga keamanan rangkaian, melindungi komponen dari pengaruh lingkungan luar, serta meningkatkan estetika agar tampilan lebih rapi dan profesional. Dengan pelindung ini, sistem dapat beroperasi secara optimal dalam berbagai kondisi kendaraan.

2.3. Rancangan Perangkat Lunak

Perangkat lunak dirancang untuk mengintegrasikan perangkat keras dengan aplikasi pengguna agar data dapat diproses dan disajikan secara efisien. Bagian ini membahas desain dan implementasi perangkat lunak, termasuk pengembangan aplikasi dan pemrograman mikrokontroler dalam sistem IoT.

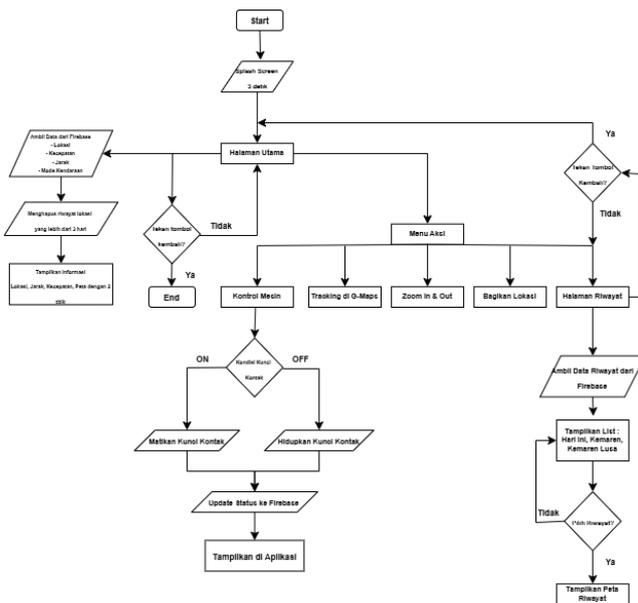
2.3.1. Desain dan Implementasi Aplikasi

Aplikasi dibuat dengan Android Studio untuk memudahkan pengguna melacak kendaraan tanpa perlu mengirim SMS atau pesan Telegram. Aplikasi memiliki tiga tampilan utama: halaman pembuka, halaman utama, dan halaman riwayat lokasi. Halaman pembuka menampilkan logo selama 3 detik sebelum ke halaman utama, yang terdiri dari *header* (informasi alamat dan kecepatan kendaraan), *middle* (peta Google Maps yang menunjukkan posisi HP dan kendaraan), serta *footer* (fitur pelacakan, berbagi lokasi, mematikan kunci kontak, dan akses riwayat lokasi). Halaman riwayat lokasi menyimpan data perjalanan kendaraan selama 3 hari terakhir. Desain Tampilan Aplikasi ditunjukkan pada Gambar 6



Gambar 6. Desain Tampilan Aplikasi

Sebagai kelanjutan dari desain tampilan aplikasi, rancangan alur kerja aplikasi digambarkan dalam bentuk *flowchart*. *Flowchart* pada Gambar 7 digunakan untuk memetakan proses operasional aplikasi, mulai dari input pengguna hingga output berupa informasi yang disajikan. Dengan *flowchart*, alur logika aplikasi dapat divisualisasikan secara sistematis untuk memastikan setiap fitur berfungsi sesuai tujuan[20].

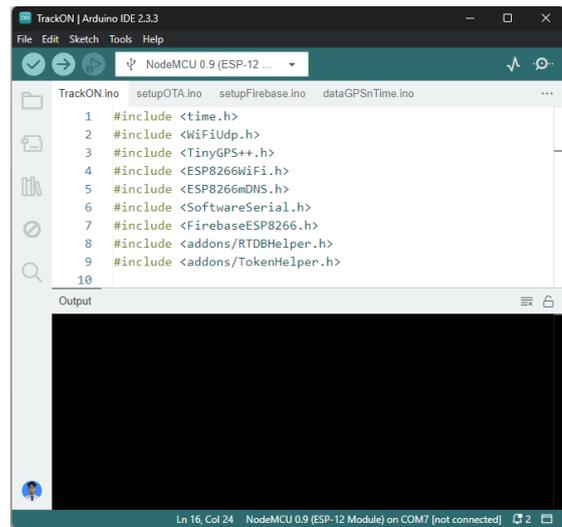


Gambar 7. Flowchat Aplikasi

Desain tampilan dan alur kerja menjadi dasar implementasi aplikasi ini. Dengan rancangan yang sistematis, aplikasi ini diharapkan dapat berjalan sesuai dengan kebutuhan pengguna, memberika kemudahan dalam pemantauan kendaraan, serta menyajikan informasi secara akurat dan *real-time*.

2.3.2. Desain dan Implementasi Program NodeMCU ESP8266

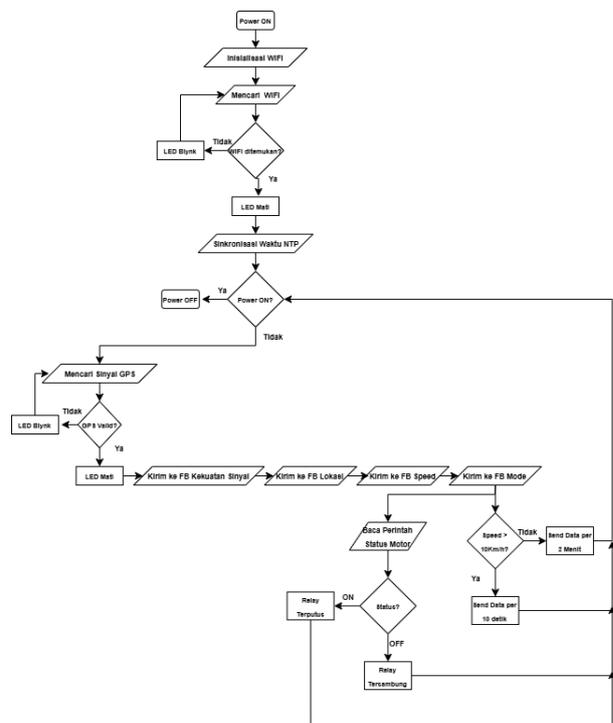
Program NodeMCU ESP8266 merupakan serangkaian instruksi yang dirancang untuk mengontrol sistem utama pada alat ini. Pemrograman dilakukan menggunakan perangkat lunak Arduino IDE dengan bahasa pemrograman C++ menggunakan pustaka-pustaka sebagaimana pada Gambar 8.



Gambar 8. Pustaka yang digunakan

Dalam proses pembuatannya, beberapa pustaka khusus digunakan untuk mengeksekusi perintah-perintah yang ada pada program. Semua pustaka tersebut diimplementasikan sesuai dengan fungsinya masing-masing untuk mendukung kebutuhan sistem secara keseluruhan.

Untuk menggambarkan alur kerja dari sistem ini dari sisi pemrograman, digunakan *flowchart* yang menjelaskan algoritma kerja sistem ini secara sederhana. *Flowchart* pada Gambar 9 mempermudah pemahaman logika program tanpa perlu membaca kode program secara keseluruhan.



Gambar 9. Flowchart Program ESP8266

Pemrograman dan alur kerja dalam bentuk *flowchart* menjadi dasar penerapan kode pada perangkat ESP8266. Tahap ini memastikan bahwa sistem berfungsi sesuai rancangan dan mampu mendukung kegiatan operasional alat secara optimal.

2.4. Metode Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi akurasi pelacakan lokasi, fitur keamanan, kemampuan penyimpanan riwayat lokasi, dan kemampuan pengiriman data secara *real-time*.

2.4.1. Pengujian Akurasi Pelacakan

Pengujian ini bertujuan untuk mengukur tingkat akurasi lokasi yang dihasilkan perangkat dibandingkan dengan koordinat aktual. Data lokasi perangkat diuji di beberapa titik yang dipilih. Koordinat dari perangkat dicatat secara berulang, kemudian dibandingkan dengan data dari GPS pada *smartphone* sebagai referensi.

Penghitungan jarak antara data perangkat dan data referensi, digunakan rumus Haversine. Rumus ini memperhitungkan kelengkungan bumi dan menghitung jarak antara dua koordinat sebagai berikut [21]:

$$\Delta\phi = \text{latitude2} - \text{latitude1} \quad (1)$$

$$\Delta\lambda = \text{longitude2} - \text{longitude1} \quad (2)$$

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (3)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (4)$$

$$d = R \cdot c \quad (5)$$

Dalam perhitungan ini, ϕ dan λ masing-masing adalah latitude dan longitude dalam radian, sementara r adalah jari-jari bumi sebesar 6371 km. Nilai *error* dihitung sebagai jarak rata-rata antara data perangkat dan data referensi.

2.4.2. Pengujian Fitur Keamanan

Pengujian ini dilakukan untuk mengevaluasi kemampuan perangkat dalam merespons perintah dalam hal keamanan. Pengujian dilakukan untuk mendapatkan waktu respon sejak perintah dikirim hingga perintah dieksekusi, serta konsistensi dalam merespon perintah selama 10 kali percobaan.

2.4.3. Pengujian Fitur Riwayat Lokasi

Pengujian ini bertujuan untuk memastikan bahwa sistem dapat mencatat dan mengirimkan riwayat lokasi dengan

akurat ke *database*. Sistem diuji dengan mencatat data lokasi selama perjalanan sejauh rute tertentu. Koordinat lokasi dicatat dengan *interval* tertentu berdasarkan kecepatan perpindahan kendaraan.

Hasil pengujian ini mencakup akurasi pencatatan data lokasi, jumlah data yang berhasil disimpan selama pengujian, serta keandalan sistem dalam menyimpan riwayat lokasi secara kontinu selama periode waktu tertentu.

2.4.4. Pengujian Kemampuan Real-Time

Pengujian ini bertujuan untuk mengevaluasi kemampuan sistem dalam mengirimkan data lokasi secara *real-time* ke aplikasi *mobile*. Sistem diuji dengan mengirimkan data lokasi dari perangkat dengan *interval* tertentu berdasarkan kecepatan perpindahan kendaraan. Data yang diterima akan dibandingkan dengan data yang dihasilkan oleh perangkat untuk memastikan tidak ada jeda waktu yang signifikan. Parameter yang diukur meliputi waktu jeda (latensi) antara pengambilan data oleh perangkat hingga data diterima di aplikasi. Sistem dianggap memenuhi kriteria *real-time* jika latensi rendah.

3. Hasil dan Pembahasan

3.1. Perangkat Keras

Perancangan dan perakitan perangkat keras dalam penelitian ini dilakukan dengan mempertimbangkan aspek fungsionalitas dan efisiensi sistem. Setiap komponen dipilih dan disusun agar dapat bekerja secara optimal dalam mendukung pengambilan serta pengiriman data. Salah satu aspek penting yang diperhatikan adalah konektivitas perangkat terhadap jaringan untuk memastikan kelancaran komunikasi data. Hasil akhir perangkat keras ditunjukkan pada Gambar 10



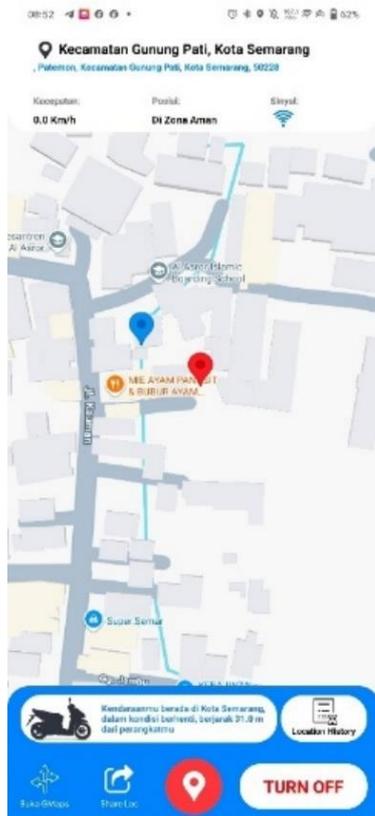
Gambar 10. Hasil akhir perangkat keras

Gambar 10 merupakan hasil akhir dari perangkat keras yang sudah berhasil diselesaikan. Pada perangkat ini, keseluruhan komponen elektronik dan sensor telah dirangkai menjadi satu untuk mendukung fungsi sistem. Penelitian sebelumnya mencatat bahwa perangkat membutuhkan jaringan WiFi eksternal untuk komunikasi data sehingga harus menyediakan modem tambahan agar alat dapat berfungsi dengan optimal[13]. Dalam penelitian ini, keterbatasan tersebut diatasi dengan mengintegrasikan modem WiFi GSM langsung ke dalam perangkat. Modem

ini memungkinkan alat untuk mengirimkan data ke server secara *real-time* tanpa memerlukan modem eksternal. Perangkat keras ini bertanggung jawab mendeteksi dan memperoleh koordinat lokasi lalu mengirimkan ke database, serta mengeksekusi perintah yang dikirimkan melalui perangkat lunak oleh pengguna.

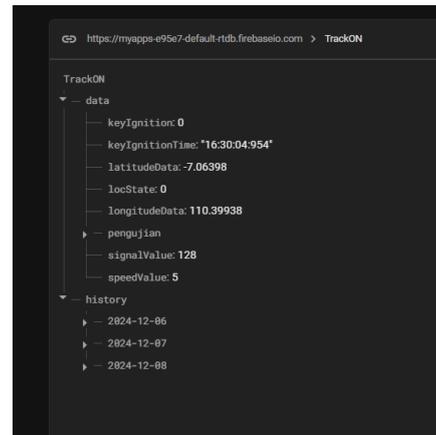
3.2. Perangkat Lunak

Aplikasi yang dikembangkan berperan sebagai antarmuka pengguna untuk mengelola dan memantau data lokasi yang diperoleh dari perangkat keras. Pada penelitian sebelumnya diperlukan sebuah metode baru yang memungkinkan pengguna melihat posisi kendaraan secara *real-time* langsung di aplikasi, tanpa harus membuka Google Maps secara terpisah[6]. Keterbatasan ini mengurangi efisiensi dan kenyamanan pengguna dalam memantau pergerakan kendaraan. Hasil akhir perangkat lunak ditunjukkan pada Gambar 11



Gambar 11. Hasil akhir perangkat lunak

Dalam penelitian ini, keterbatasan tersebut diatasi dengan mengembangkan aplikasi mobile yang terintegrasi dengan Google Maps API. Aplikasi ini seperti yang terlihat pada Gambar 11, dapat menampilkan posisi pengguna pada peta secara *real-time*, mencatat dan menyimpan riwayat pergerakan selama tiga hari, serta memberikan perintah ke perangkat keras. Desain antarmuka yang intuitif memudahkan pengguna dalam mengoperasikan sistem ini. Real-Time Database Firebase ditunjukkan pada Gambar 12



Gambar 12. Real-Time Database Firebase

Selain itu, database Firebase yang digunakan dalam sistem ini berfungsi untuk menyimpan data lokasi pengguna secara aman dan terorganisir. Firebase memungkinkan sistem untuk menyimpan serta mengakses data lokasi secara *real-time*, memastikan bahwa informasi yang diperoleh tetap akurat dan dapat diandalkan. Contoh tampilan data yang tersimpan di Firebase ditunjukkan pada Gambar 12, yang memperlihatkan struktur penyimpanan dan format data yang digunakan dalam sistem ini.

3.3. Pengujian Koordinat Lokasi

Pengujian koordinat lokasi dilakukan untuk mengevaluasi tingkat akurasi perangkat dalam menentukan posisi pada berbagai titik uji. Dalam pengujian ini, hasil koordinat dari perangkat dibandingkan dengan koordinat referensi yang diperoleh menggunakan aplikasi GPS Coordinates sebagai instrumen pembanding. Dengan membandingkan kedua hasil tersebut, dapat diketahui sejauh mana perbedaan yang terjadi. Nilai *error* dihitung untuk memberikan gambaran tentang tingkat ketelitian sistem dalam mengukur lokasi. Hasil pengujian di 10 titik lokasi ditunjukkan pada Tabel 1

Tabel 1. Hasil pengujian di 10 titik lokasi

No	Alat		Instrumen Pembanding		Error (meter)
	Longitude	Latitude	Longitude	Latitude	
1.	-7,05128	110,39941	-7,05176	110,40030	111,78
2.	-7,04966	110,39722	-7,05003	110,39799	94,40
3.	-7,04979	110,39344	-7,04968	110,39378	39,46
4.	-7,0509	110,39301	-7,05077	110,39327	32,12
5.	-7,05147	110,39239	-7,05152	110,39256	19,56
6.	-7,05097	110,39098	-7,05080	110,39091	20,42
7.	-7,04975	110,39014	-7,04889	110,39010	95,72
8.	-7,0492	110,39141	-7,04920	110,39140	1,10
9.	-7,05011	110,39849	-7,05010	110,39827	24,30
10.	-7,05154	110,40067	-7,05154	110,40053	15,45

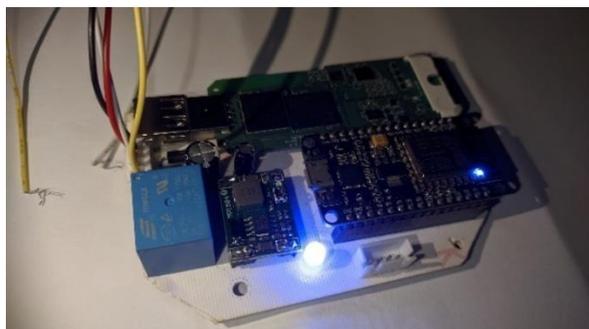
Tabel 1 menunjukkan hasil pengujian koordinat lokasi pada 10 titik berbeda. Dari data diatas, terlihat perbedaan

antara koordinat hasil uji dan koordinat referensi. Perbedaan ini dinyatakan dalam nilai *error* yang dihitung dalam satuan meter.

Secara umum, nilai *error* yang ditampilkan cukup bervariasi, mulai dari yang terkecil yaitu 1,10 meter pada titik ke-8, hingga yang terbesar 111,78 meter pada titik pertama. Hal ini mengindikasikan bahwa akurasi pengukuran oleh alat masih perlu ditingkatkan, terutama pada titik-titik tertentu yang menunjukkan *error* yang cukup besar.

3.4. Pengujian Fitur Keamanan Kendaraan

Pada pengujian ini dilakukan dengan menguji konsistensi dan respon waktu pada saat tombol pada aplikasi dioperasikan hingga alat bekerja. Pada pengujian, relay aktif ditandai dengan LED yang menyala. Pengujian ini menggunakan metode uji fungsi, di mana sistem diuji berdasarkan respons aktualnya terhadap skenario yang telah dirancang. Karena aspek yang diuji berkaitan dengan performa internal sistem, pengujian ini tidak memerlukan instrumen pembanding eksternal, melainkan dianalisis berdasarkan hasil observasi langsung terhadap keluaran sistem.



Gambar 13. LED indikator menyala

Pada Gambar 13 relay aktif dan akan memutus arus yang mengalir ke kunci kontak, yang mengakibatkan motor tidak bisa dinyalakan. Hasil pengujian disajikan dalam Tabel 2.

Tabel 2. Hasil pengujian fitur keamanan

No	Waktu Terkirim	Waktu Diterima	Relay	Respon Aplikasi	latensi (ms)
1.	10:04:50:757	10:04:51:025	ON	Motor Telah Dimatikan	258
2.	10:05:39:747	10:05:40:101	OFF	Motor Telah Dinyalakan	354
3.	10:08:22:027	10:08:22:214	ON	Motor Telah Dimatikan	187
4.	10:08:48:856	10:08:48:994	OFF	Motor Telah Dinyalakan	138
5.	10:10:57:640	10:10:57:949	ON	Motor Telah Dimatikan	309
6.	10:13:48:530	10:13:48:774	OFF	Motor Telah Dinyalakan	244
7.	10:15:54:739	10:15:54:894	ON	Motor Telah Dimatikan	155

No	Waktu Terkirim	Waktu Diterima	Relay	Respon Aplikasi	latensi (ms)
8.	10:16:57:279	10:16:57:413	OFF	Motor Telah Dinyalakan	134
9.	10:18:09:212	10:18:09:457	ON	Motor Telah Dimatikan	245
10.	10:19:23:770	10:19:23:963	OFF	Motor Telah Dinyalakan	193

Berdasarkan hasil pengujian tersebut, didapatkan konsistensi 100% dari 10 kali percobaan, menunjukkan bahwa sistem bekerja secara andal dalam setiap skenario pengujian. Respon waktu yang tercatat juga menunjukkan performa yang baik, dengan respon tercepat 134 ms dan respon terlama 354 ms. Waktu respon ini cukup singkat untuk memastikan bahwa sistem dapat diaktifkan tanpa penundaan yang signifikan. Dengan kata lain, fitur keamanan ini mampu beroperasi dengan akurasi tinggi dan kecepatan yang memadai dalam situasi nyata. Namun, dalam implementasi lebih lanjut, perlu dipertimbangkan pengujian dalam kondisi lingkungan yang lebih variatif, seperti perbedaan suhu atau gangguan sinyal, guna memastikan sistem tetap dapat berfungsi secara optimal dalam berbagai kondisi operasional.

3.5. Pengujian Fitur Riwayat Lokasi

Pengujian dilakukan untuk memastikan sistem mencatat data lokasi selama perjalanan sejauh rute tertentu dan mengirimkan riwayat lokasi dengan akurat ke database. Untuk memverifikasi keakuratan data yang tercatat, hasil dari sistem dibandingkan dengan riwayat lokasi yang diperoleh menggunakan aplikasi Google Maps sebagai instrumen pembanding. Aplikasi ini digunakan untuk mengetahui riwayat lokasi per titik yang dilalui selama perjalanan.

Tabel 3. Hasil pengujian fitur riwayat lokasi

No	Tanggal	Alamat Awal	Alamat Akhir	Jumlah Titik
1.	06/12/2024	W9PX+8PJ, Patemon, Gunung Pati, Semarang City, Central Java 50228, Indonesia	Fakultas Teknik UNNES, Sekaran, Kec. Gn. Pati, Kota Semarang, Jawa Tengah, Indonesia	32
2.	07/12/2024	X92V+7W2, Sekaran, Gunung Pati, Semarang City, Central Java 50229, Indonesia	FT UNNES, Sekaran, Kec. Gn. Pati, Kota Semarang, Jawa Tengah, Indonesia	21
3.	08/12/2024	FT UNNES, Sekaran, Kec. Gn. Pati, Kota Semarang, Jawa Tengah, Indonesia	FT UNNES, Sekaran, Kec. Gn. Pati, Kota Semarang, Jawa Tengah	10

Tabel 3 menunjukkan hasil pengujian fitur riwayat lokasi pada sistem. Dari tabel tersebut, dapat dilihat informasi

tanggal pengujian, alamat awal, alamat akhir, serta jumlah titik lokasi yang berhasil tercatat.



Gambar 15. Tampilan riwayat lokasi (07/12/2024)

Berdasarkan Gambar 15 menunjukkan bahwa riwayat lokasi tercatat dengan baik dan menyimpannya ke dalam *database* secara kontinu selama periode waktu pengujian. Jumlah titik lokasi yang berhasil tercatat juga menunjukkan keandalan sistem dalam melakukan pencatatan riwayat lokasi.

3.6. Pengujian Kemampuan Real-time

Pada pengujian kemampuan *real-time* sistem ini, dilakukan pengukuran latensi antara waktu pengiriman data oleh alat dan yang tercatat oleh sistem. Hasil pengujian menunjukkan nilai latensi dalam milidetik (*ms*) dengan hasil terdapat pada Tabel 4. Pengujian ini dilakukan dengan pendekatan eksperimental, di mana data latensi diukur langsung dari sistem tanpa memerlukan instrumen pembanding eksternal. Evaluasi dilakukan dengan membandingkan hasil yang diperoleh terhadap standar keterlambatan yang dapat diterima dalam sistem *real-time*.

Tabel 4. Hasil pengujian kemampuan real-time

No	Waktu Terkirim (HH:MM:SS:MMM)	Waktu Diterima (HH:MM:SS:MMM)	Latensi (ms)
1.	08:12:32.653	08:12:34.999	2346
2.	08:13:02.670	08:13:04.999	2329
3.	08:13:32.592	08:13:34.000	1408
4.	08:14:02.558	08:14:04.000	1442
5.	08:14:32.799	08:14:34.000	1201
6.	08:15:02.580	08:15:04.000	1420
7.	08:15:32.597	08:15:34.000	1403
8.	08:16:02.568	08:16:04.000	1432
9.	08:16:32.542	08:16:34.000	1458
10.	08:17:02.739	08:17:04.000	1261

Data menunjukkan fluktuasi latensi yang signifikan, dengan nilai tertinggi 2346 ms, terendah 1201 ms, dan rata-rata 1423 ms. Meskipun masih dalam batas toleransi untuk pelacakan posisi, sistem belum sepenuhnya *real-time*. Keterlambatan ini kemungkinan dipengaruhi oleh kestabilan jaringan, beban *server*, atau efisiensi algoritma pengiriman data. Untuk meningkatkan performa, perlu optimasi seperti pengurangan ukuran paket data, peningkatan efisiensi koneksi, atau penggunaan protokol yang lebih responsif seperti MQTT dibandingkan HTTP, agar latensi lebih stabil dan mendekati *real-time*.

4. Kesimpulan

Sistem pelacakan kendaraan yang dikembangkan berhasil mencapai tujuan utama, yaitu pelacakan *real-time*, pemantauan riwayat lokasi, dan peningkatan keamanan kendaraan melalui aplikasi mobile yang terintegrasi dengan Google Maps API dan *database* Firebase. Hasil pengujian menunjukkan bahwa sistem mampu mencatat lokasi secara akurat dengan tingkat *error* yang bervariasi, mulai dari 1,10 meter hingga 111,78 meter, yang menunjukkan bahwa akurasi masih dapat ditingkatkan. Fitur keamanan kendaraan terbukti andal dengan tingkat keberhasilan 100% dalam 10 kali pengujian serta waktu respon yang sangat cepat, berkisar antara 134 ms hingga 354 ms. Namun, pada pengujian *real-time*, sistem mencatat latensi rata-rata 1423 ms, dengan fluktuasi antara 1201 ms hingga 2346 ms, yang mengindikasikan bahwa sistem belum sepenuhnya *real-time* tetapi masih dalam batas toleransi untuk aplikasi pelacakan posisi. Variasi latensi ini kemungkinan disebabkan oleh faktor jaringan dan metode transmisi data, sehingga diperlukan optimasi lebih lanjut, seperti penggunaan GPS dengan presisi lebih tinggi dan protokol komunikasi yang lebih efisien untuk meningkatkan keandalan dan responsivitas sistem dalam kondisi operasional yang lebih luas.

Referensi

- [1] Badan Pusat Statistik, “Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis (Unit). Diakses dari: <https://www.bps.go.id/id/statistics-able/2/NTcjMg==/perkembangan-jumlah-kendaraan-bermotor-menurut-jenis--unit-.html>,” BPS
- [2] F. S. Pratiwi, “Data Jumlah Kasus Pencurian Kendaraan Bermotor di Indonesia pada 2022. Diakses dari: <https://dataindonesia.id/varia/detail/data-jumlah-kasus-pencurian-kendaraan-bermotor-di-indonesia-pada-2022>,” DataIndonesia.id.
- [3] I. Yustiana and M. S. Mulya, “Keamanan Kendaraan untuk Melacak Sepeda Motor yang Hilang dengan menggunakan GPS Berbasis Smartphone,” *MEANS(Media Informasi Analisa dan Sistem)*, vol. 6, no. Keamanan Kendaraan untuk Melacak Sepeda Motor yang Hilang dengan menggunakan GPS Berbasis Smartphone, pp. 225–228, Dec. 2021.
- [4] A. Bessarab, T. Гиріна, O. Sytnyk, N. Kodatska, O. Yatchuk, and L. Ponomarenko, “THE MODERN TRANSFORMATION OF INTERNET COMMUNICATIONS,” *J Theor Appl Inf Technol*, vol. 100, pp. 4710–4722, Aug. 2022.
- [5] A. C. Rahman, W. A. Arimbawa, and A. H. Jatmika, “IMPLEMENTASI INTERNET OF THINGS PADA SISTEM INFORMASI PELACAKAN KENDARAAN BERMOTOR MENGGUNAKAN GPS BERBASIS WEB (Implementation of Internet of Things on Web-Based Motor Vehicle Tracking Information System Using GPS),” Mar. 2019. [Online]. Available: <http://jtika.if.unram.ac.id/index.php/JTIKA/>
- [6] Risaldi, Mukramin, R. Suppa, and V. I. Wahyuni, “RANCANG BANGUN ALAT PELACAK POSISI KENDARAAN BERBASIS IoT,” *JUTINDA (Jurnal Teknik Informatika Unanda)*, vol. 3, pp. 29–33, May 2024, Accessed: Nov. 19, 2024. [Online]. Available: <https://ojs.unanda.ac.id/index.php/jutinda>
- [7] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, “A Review and State of Art of Internet of Things (IoT),” May 01, 2022, *Springer Science and Business Media B.V.* doi: 10.1007/s11831-021-09622-6.
- [8] M. M. Islam, S. Nooruddin, F. Karray, and G. Muhammad, “Internet of Things: Device Capabilities, Architectures, Protocols, and Smart Applications in Healthcare Domain,” *IEEE Internet Things J*, vol. 10, no. 4, pp. 3611–3641, Feb. 2023, doi: 10.1109/JIOT.2022.3228795.
- [9] K. Liu, Y. R. Bi, and D. Liu, “Internet of Things based acquisition system of industrial intelligent bar code for smart city applications,” *Comput Commun*, vol. 150, pp. 325–333, Jan. 2020, doi: 10.1016/J.COMCOM.2019.11.044.
- [10] P. Upadhyay and Prof. D. Upadhyay, “Internet of things - A Survey,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 417–438, May 2021, doi: 10.32628/cseit217394.
- [11] M. Ashfaq and S. Nur, “IoT Sensor Networks-Orchestrating Connectivity, Efficiency, and Intelligence Across Diverse Domains,” *International Journal of Innovative Research in Computer Science and Technology*, vol. 12, no. 3, pp. 154–161, May 2024, doi: 10.55524/ijirest.2024.12.3.26.
- [12] J. Zhang, Y. Ye, C. Hu, and B. Li, “Architecture design and demand analysis on application layer of standard system for ubiquitous power Internet of Things,” *Global Energy Interconnection*, vol. 4, no. 3, pp. 304–314, Jun. 2021, doi: 10.1016/j.gloi.2021.07.001.
- [13] M. Arif Budiman, A. Zatulo Harefa, and D. Virgian Shaka, *PERANCANGAN SISTEM PELACAK GPS DAN PENGENDALI KENDARAAN JARAK JAUH BERBASIS ARDUINO. SENDIU*, 2020.
- [14] D. H. Wicaksono, D. Djuniadi, and E. Apriaskar, “Monitoring Sistem Pembangkit Listrik Tenaga Angin Berbasis Internet of Things,” *Jurnal Teknologi Elektro*, vol. 14, no. 2, p. 118, Jul. 2023, doi: 10.22441/jte.2023.v14i2.010.
- [15] M. Wijayanti, “PROTOTYPE SMART HOME DENGAN NODEMCU ESP8266 BERBASIS IOT,” *JUIT*, vol. 1, no. 2, May 2022.
- [16] P. Kanani and Dr. M. Padole, *Real-time Location Tracker for Critical Health Patient using Arduino, GPS Neo6m and GSM Sim800L in Health Care*. IEEE, 2020.
- [17] Y. K. Verma, M. Jagadeesh, P. M. Kumar, H. I. Younis, and M. S. Adhikari, “Design of Real-Time GPS Tracker Using ESP-8266 With Blynk Application,” in *2023 International Conference on IoT, Communication and Automation Technology, ICICAT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICICAT57735.2023.10263731.
- [18] F. Juliansyah and R. Rachmatika, “OKTAL : Jurnal Ilmu Komputer dan Science ANALISIS PENGARUH INTERFERENSI WIFI TERHADAP QUALITY OF SERVICE (QOS) PADA MODEM WIRELESS HUAWEI EG8145V5 DENGAN METODE ACTION RESEARCH,” vol. 2, no. 11, 2023.
- [19] S. Eka, M. Y. Abdussalam, Z. A. Husniah, A. P. Lestari, and R. Z. Hana, “Prototype Design of a Fishing Boat Safety Monitoring System Using LoRa and Microsensor Devices,” *JMECS (Journal of Measurements, Electronics, Communications, and Systems)*, vol. 10, pp. 77–85, Dec. 2023.
- [20] W. Ochoa, F. Larrinaga, and A. Pérez, “Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches,” *Future Generation Computer Systems*, vol. 145, pp. 38–55, Aug. 2023, doi: 10.1016/j.future.2023.03.017.
- [21] E. Budiayanti, F. Psikologi, U. Gunadarma, J. Margonda Raya, and J. Barat, “IMPLEMENTASI METODE HAVERSINE FORMULA PADA SISTEM INFORMASI GEOGRAFIS RUMAH KOST DAERAH TANGERANG SELATAN,” vol. 1, no. 1, 2022.